Oracle Software

System Requirement Specification

Submitted to:

Submitted by:



Version: 1.1

Date: September 25, 2024

Table of Contents

[Table of Contents 2](#_Toc1817019623)

[1 Construction of a cst image 3](#_Toc1259160075)

[2 Updating the SRK binaries into YOCTO build 5](#_Toc1930530418)

# Construction of a cst image

Download the NXP Code Signing Tool (CST) from below link:

[https://www.nxp.com/webapp/sps/download/license.jsp?colCode=IMX\_CST\_TOOL](")

**Step 1:** Unpack the downloaded archive:

$ cd ~

$ tar xf ~/Downloads/cst-3.1.0.tar.gz

**Step 2:**

Generate Public Key Infrastructure (PKI) tree

$ cd ~/<your-path>/cst-3.1.0/keys

**Step 3:**

Create a text file called "serial", which contains 8 digits. It will be used for the certificate serial numbers.

For example:

$ echo 1248163E > serial

**Step 4:**

Create a text file called "key\_pass.txt", which contains two lines of a password repeated twice.

This password will be used to protect the generated private keys.

All private keys in the PKI tree are in PKCS #8 format will be protected by the same password.

For example:

$ echo oracle\_password > key\_pass.txt

$ echo oracle\_password >> key\_pass.txt

**Step 5:**

Now, to generate the PKI tree, run the following:

$ ./ahab\_pki\_tree.sh

And complete the interactive questions. For example:

Do you want to use an existing CA key (y/n)?: n

Do you want to use Elliptic Curve Cryptography (y/n)?: n

Enter key length in bits for PKI tree: 4096

Enter the digest algorithm to use: sha384

Enter PKI tree duration (years): 20

Do you want the SRK certificates to have the CA flag set? (y/n)?: n

**Step 6:**

Generate Super Root Key (SRK) table

$ cd ../crts/

$ ../linux64/bin/srktool -a -s sha384 -t SRK1234table.bin -e SRK1234fuse.bin -f 1 -c SRK1\_sha384\_4096\_65537\_v3\_usr\_crt.pem,SRK2\_sha384\_4096\_65537\_v3\_usr\_crt.pem,SRK3\_sha384\_4096\_65537\_v3\_usr\_crt.pem,SRK4\_sha384\_4096\_65537\_v3\_usr\_crt.pem

$ ll SRK1234\*

-rw-rw-r-- 1 nate nate 64 Sep 15 14:47 SRK1234fuse.bin

-rw-rw-r-- 1 nate nate 2112 Sep 15 14:47 SRK1234table.bin

**Step 7:**

Prepare the Image Signing Configuration File

* Create the configuration file (ahab\_image.csf) to specify the signing process.
* In this file, include the paths for SRK1234table.bin and SRK1234fuse.bin, and define the image signing requirements.

$ touch ahab\_image.csf

$ vi ahab\_image.csf

**Example ahab\_image.csf:**

|  |
| --- |
| **[Header]**  **Version = 4.3**  **Hash Algorithm = sha256**  **Engine Configuration = 0**  **[Install SRK]**  **File = "SRK1234table.bin"**  **Source index = 0**  **[Authenticate Data]**  **Verification Index = 0**  **Blocks = <boot\_image\_address> <size\_of\_image>** |

**Step 8:**

Sign the Image and Generate the AHAB Container

$ ../linux64/bin/cst -i ahab\_image.csf -o ahab\_container.bin

# Updating the SRK binaries into YOCTO build

Move the generated ahab\_container.bin to your Yocto deployment directory so it can be included in your Yocto images.

Copy to the below mentioned path:

~/stryker\_ws/build-imx93-evk/tmp/deploy/images/imx93evk/ahab\_container.bin